

## PERANCANGAN APLIKASI DASHBOARD WMS BERBASIS WEB SERVICE DENGAN MENGGUNAKAN TEKNOLOGI .NET WEB SERVICE DAN PHP

Ryan Permana<sup>1</sup>, Didiék S Wiyono<sup>1</sup>, Nanang Ismail<sup>2</sup>

<sup>1</sup>Prodi Teknik Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Sebelas Maret Surakarta

<sup>2</sup>Jurusan Teknik Elektro, Fakultas Sains dan Teknologi,  
Universitas Islam Negeri Sunan Gunung Djati Bandung

### Abstract

*Warehouse Management System needs an item inventory, configuration saving method of items, integration among warehouses, and warehouse finance control. All of the aspects must be well arranged, in order to avoid warehouses's work performance decrease that can affect to supply chain profit. To avoid the supply chain from losses, a warehouses need a system that can help to maintain and manage all of warehouse's variables. A system that can implemented is WMS Dashboard. WMS Dashboard is an application that can show the warehouse's data realtime and with high accuration.*

*WMS Dashboard's services are made by using ASP.NET Web Service Technology which use SOAP protocol and WSDL interface so it support multi platform system to system communication which using HTTP protocol. The database of WMS Dashboard is made by using MS SQL Server DBMS which has fully compatibility with other applications that made by .NET Framework. Object oriented PHP is need to made the application for client side.*

*WSDL of WMS dashboard application consist of 56 services. The services has been integrated with WMS Dashboard Application which based on PHP, so it would be accessed easily, and it can be used to help maintain and manage the Warehouse System variables.*

*Keywords: Asp.net web service, Ms SQL, object oriented PHP, WMS dashboard*

### A. Pendahuluan

Gudang sebagai sarana penyimpanan stok sumber daya merupakan bagian yang memiliki peran penting dalam efektivitas sebuah rantai pasok. Dengan fungsinya sebagai penyimpanan stok maka gudang memerlukan sebuah sistem manajemen untuk mengorganisir kegiatan- kegiatan yang ada dalam gudang.

Dalam me-manage sebuah gudang terdapat banyak hal yang perlu dikelola dengan baik, seperti menentukan metode penyimpanan stok sesuai dengan kriteria yang dibutuhkan stok tersebut, pengintegrasian antar gudang dalam satu wilayah dengan wilayah lainnya, pengelolaan keuangan seperti pengelolaan uang masuk dan keluar serta pengelolaan keuntungan, dan lain sebagainya. Pengelolaan hal-hal tersebut

membutuhkan ketelitian dan keakuratan, serta kestabilan dalam pemantauannya, sehingga dibutuhkan sebuah sistem yang mampu diandalkan untuk melakukannya dengan tiga spesifikasi tersebut (teliti, akurat dan stabil). Salah satu sistem yang diajukan untuk membantu masalah yang terjadi dalam manajemen pergudangan adalah Warehouse Management System *Dashboard* (WMS-Dashboard). WMS *Dashboard* adalah sebuah aplikasi yang dapat menampilkan data dari variable-variable yang ada dalam gudang secara *realtime* dengan tingkat keakuratan yang tinggi.

Sistem pergudangan perusahaan-perusahaan di Indonesia yang kebanyakan masih menggunakan metode manual dan belum saling terintegrasi dapat menimbulkan peluang untuk terjadinya *human error* yang dapat berakibat pada ruginya perusahaan.

WMS-Dashboard diharapkan mampu menangani masalah dalam pengelolaan penyimpanan sumber daya dari sebuah rantai pasok yang masih belum tertata dengan baik, terutama dalam masalah pemantauan variabel-variabel yang ada pada gudang seperti jumlah stok dalam gudang dan keadaan stok dalam gudang dengan tingkat keakuratan, ketelitian, kestabilan tingkat fleksibilitas yang tinggi.

## B. Tinjauan Pustaka

### 1. Pengertian Warehouse Management System

*Warehouse Management System* adalah sebuah *real-time* sistem yang memiliki tujuan utama untuk mengatur sumber daya gudang, seperti ruang, buruh, perlengkapan, aktivitas- aktivitas dan alur stok. [3]

Dengan WMS, kita dapat mengontrol proses pergerakan dan penyimpanan dengan lebih baik, pemakaian *space* gudang dengan lebih optimal, meningkatkan efektifitas proses penerimaan dan pengiriman serta mengetahui jumlah stok dengan lebih akurat setiap waktu.

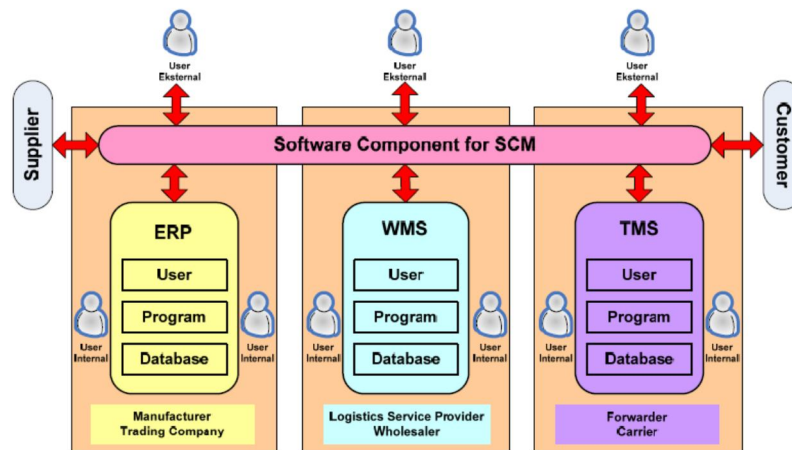
Penerapan WMS yang optimal akan memberikan keuntungan pada perusahaan (dalam kasus ini Supply Chain). Karena dengan penggunaan WMS maka akan mengoptimalkan tenaga kerja, menurangi waktu proses, mengurangi proses inventory yang tidak perlu dan akhirnya akan meningkatkan pelayanan kita kepada customer

### 2. Hubungan WMS dengan SCM

Martin Verwijmeren berpendapat bahwa manajemen lokal dari suatu arsitektur SCM dikerjakan oleh Sistem *Enterprise Resource Planning* (ERP),

*Warehouse Management System* (WMS) dan *Transportation Management System* (TMS). SCM sendiri berjalan di atas

manajemen lokal dan berfungsi sebagai *integrator* dari manajemen lokal dengan *supplier* dan *consumer*. [2]



**Gambar 1. Software Component arsitektur dalam SCM [1]**

### 3. Web Service

Web service merupakan salah satu implementasi dari teknologi XML pada proses pertukaran data (*data exchange*) antar platform yang berbeda. Definisi *web service* menurut Richards Robert. adalah [5] :

“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML seriali-zation in conjunction with other Web-related standards”.

Menurut Richards, *web service* dapat digunakan untuk berkomunikasi antara mesin satu dengan mesin yang lain melalui *interface* perantara yang umumnya berupa WSDL (Web Service Definition Language), layanan ini biasa bekerja pada protokol HTTP dengan bentuk *response* dan *request* berupa SOAP *message*.

Dalam Web Service terdapat tiga actor utama, yaitu :

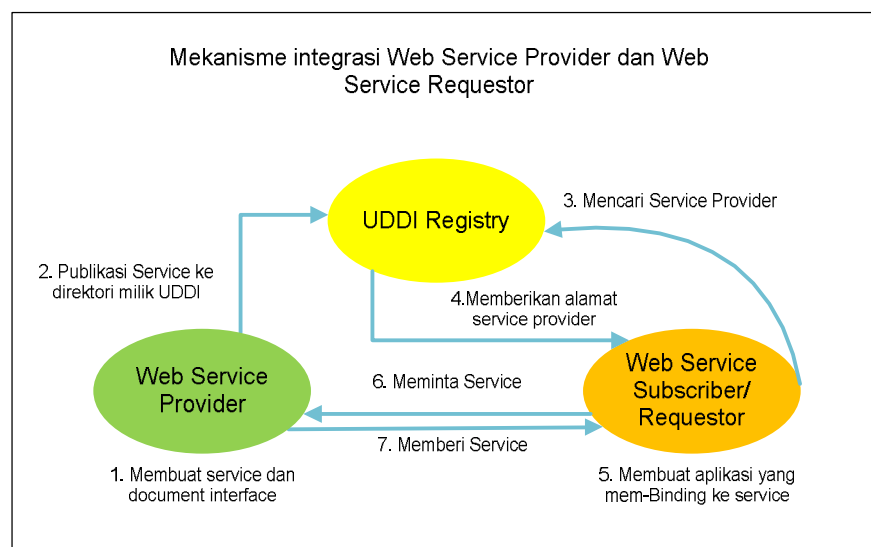
- Web Service provider → merupakan pemilik dari layanan yang akan mempublikasikannya menggunakan WSDL
- Service Registry (UDDI Registry) → merupakan pihak yang mempublikasikan layanan- layanan

milik service provider yang telah berbentuk WSDL.

- c. Web Service Requestor → merupakan web service yang akan mencari layanan yang ia butuhkan ke Service registry, kemudian Service Registry akan memberikan informasi mengenai layanan yang ia cari dalam bentuk

WSDL, kemudian Web Service Requestor akan mengambil WSDL tersebut, untuk mendapatkan cara mengakses layanan dari Web Service Provider.

Untuk lebih jelasnya, silahkan lihat ilustrasi di bawah ini :



**Gambar 2. Mekanisme Integrasi Web Service Provider dengan Web Service Requestor[1]**

#### 4. Pengertian ASP .NET Web Service

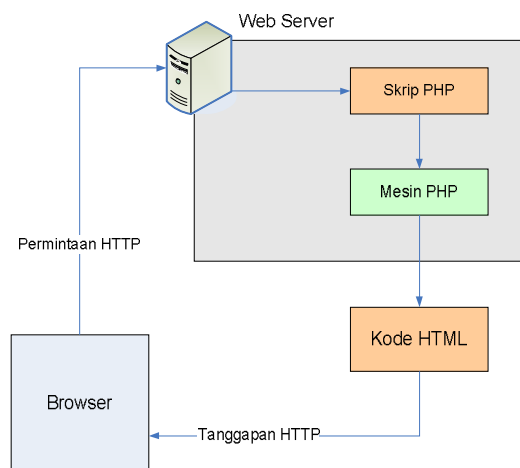
ASP.NET web services adalah sebuah teknologi web service keluaran Microsoft yang menggunakan protokol SOAP (Simple Object Access Protocol) dan dibuat menggunakan .Net Framework.[6] ASP.Net Web Service menggunakan bahasa pemrograman C# dan teknik- teknik pemrograman .Net Framework, namun meskipun menggunakan bahasa dan teknik yang

beraroma kental dengan Microsoft, service- service yang dibuat dengan .Net Framework tetap bisa dikonsumsi oleh berbagai macam platform, selama platform tersebut mendukung penggunaan protokol HTTP untuk berhubungan dengan Server. Sehingga bisa disebut juga service- service yang dibuat menggunakan .Net Framework merupakan service yang multi platform dan juga multi OS.

Web Service yang dibuat dengan menggunakan .Net Framework memiliki 2 file utama, yaitu file service yang berekstensi \*.asmx dan file kode pendukung/code behind yang berekstensi \*.cs.

## 5. Pengertian PHP

PHP (PHP: *Hypertext Preprocessor*) merupakan salah satu dari bahasa pemrograman berbasis website. PHP bersifat *server-side programming*, artinya kode PHP yang ditulis akan dieksekusi di sisi server sehingga pengunjung tidak dapat melihat *source code* dari skrip PHP yang dibangun. Adapun proses eksekusi kode PHP didalam sisi server ditunjukkan oleh gambar dibawah [4] :



## Gambar3. Proses Eksekusi kode PHP

Keunggulan PHP dibanding bahasa pemrograman web yang lain antara lain : bersifat *multi platform*, *open source*, memiliki fasilitas untuk OOP(*Object Oriented Programming*) yang merupakan teknik pemrograman yang paling handal dan banyak digunakan saat ini, bersifat gratis, memiliki dukungan API (*Application Programming Interface*) yang sangat lengkap serta didukung oleh hampir semua web hosting yang ada didunia [8].

## C. Analisis Dan Perancangan

### 1. WMS Dashboard SRS

SRS untuk aplikasi dashboard WMS ini dibagi menjadi 2 bagian yaitu SRS fungsional (SRS-WDF) yang menjabarkan kebutuhan user yang berkaitan langsung dengan fungsi utama dari aplikasi dan SRS non-fungsional (SRS-WDNF) yang menjabarkan kebutuhan user yang memanfaatkan fungsi tambahan dari aplikasi. SRS fungsional dari WMS Dashboard dijabarkan pada tabel 1 berikut ini :

**Tabel 1. SRS Fungsional**

KODE SRS	DESKRIPSI KEBUTUHAN FUNGSIONAL
<b>SCM Engine/Admin</b>	
SRS-WDF 1	Set stocks specification (Fill the stock)
SRS-WDF 2	Take stocks from warehouse
SRS-WDF 3	Get stocks information

SRS-WDF 4	Get stock alert
SRS-WDF 5	Get the source warehouse location
SRS-WDF 6	Get All Transactional Data
SRS-WDF 7	Get KPI Information
SRS-WDF 8	Get Profit Information

Sedangkan untuk SRS non fungsionalnya akan dijabarkan

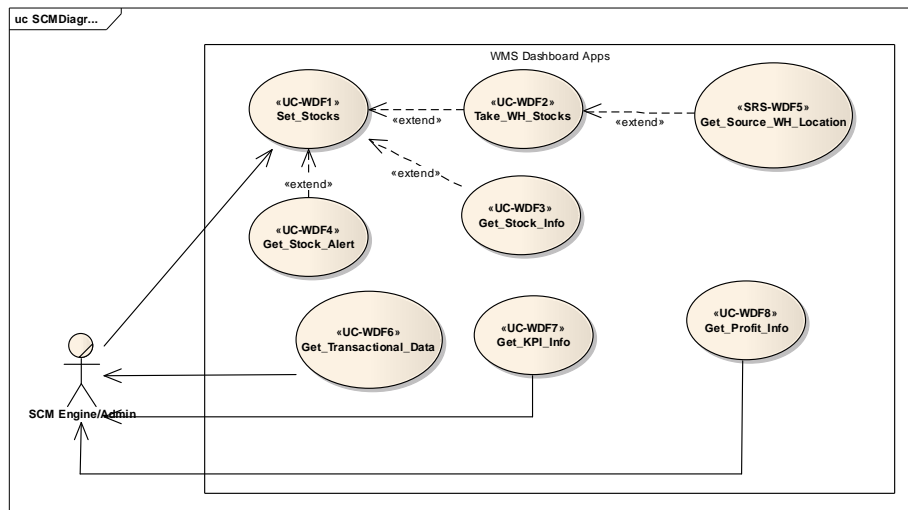
**Tabel 2. SRS Non Fungsional**

KODE SRS	DESKRIPSI KEBUTUHAN NON-FUNGSIONAL
<b>User/Owner</b>	
SRS-WDNF 1	Get User Friendly Interface
SRS-WDNF 2	Set Login/Logout
SRS-WDNF 3	Set Password
SRS-WDNF 4	Set Login Email
SRS-WDNF 5	See&Print Transactional Data
SRS-WDNF 6	See&Print KPI Information
SRS-WDNF 8	See&Print Profit Information
<b>SCMEngine/Admin</b>	
SRS-WDNF 9	Authoring Process
<b>General User</b>	
SRS-WDNF10	Get WMS Dashboard Info

**2. WMS Dashboard Use Case**

Berdasarkan SRS Fungsional dan non-Fungsional WMS Service dan berdasarkan aktor- aktor yang terlibat didalamnya, usecase diagram untuk WMS Service dapat dibagi menjadi 2 bagian

yaitu usecase fungsional untuk aktor SCM/Admin serta usecase non fungsional. Use Case pertama adalah use case untuk SCM Engine/ Admin yang akan dijelaskan pada gambar 4 berikut ini:

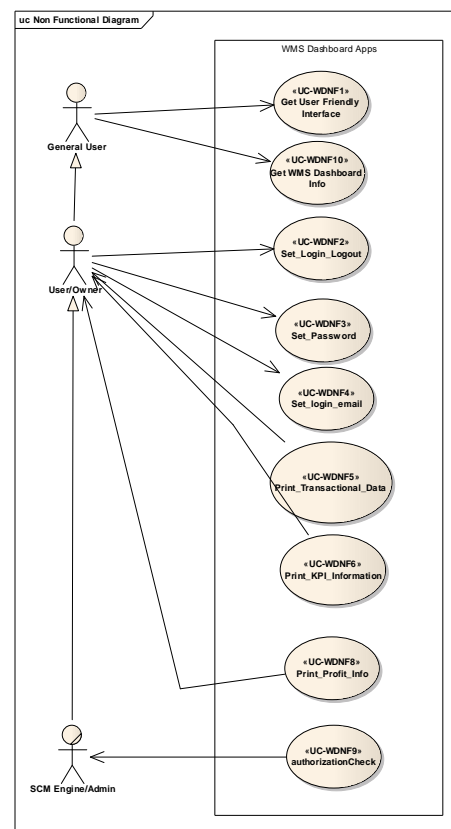


**Gambar 4. Use Case untuk SCM/Admin**

Use case diagram untuk SCM /Admin terdiri dari 8 use case, dari 8 use case terdapat 5 use case yang saling berhubungan, yaitu use case Set Stocks, Get Stock Alert, Take WH Stock, Get Stock Info dan Get Source WH Location. Kelima use case tersebut saling dihubungkan dengan relasi extend. Use case take WH Location, Get Stock info dan Get Stock Alert akan muncul/ aktif

jika use case Set Stock ada. Use case Get Source WH Location juga bergantung kepada use case Take WH Stock, karena jika use Take WH Stock, maka use case Get Source WH Location juga tidak ada.

Sedangkan untuk use case yang kedua adalah use case non- fungsional, seperti yang dijelaskan pada gambar 5 dibawah ini :

**Gambar 5. Use Case non- fungsional**

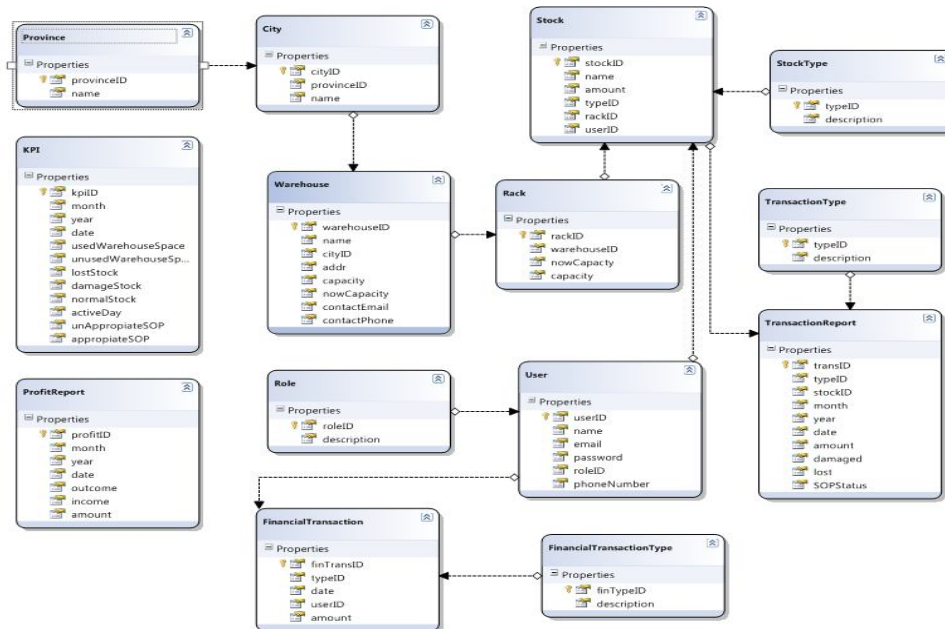
### 3. Skema Diagram

Skema diagram dari WMS Dashboard berisi diagram database milik WMS Service yang didalamnya dijelaskan

mengenai hubungan antar tabel- tabel yang ada dalam database WMS Service. Dalam service WMS Dashboard terdapat 14 tabel, dimana 12 tabel saling

terkoneksi satu sama lain, dan dua tabel lainnya merupakan tabel trigger yang akan menampung data- data dari 12 tabel

lainnya. Dibawah ini adalah gambar 6 yang menunjukkan skema diagram dari WMS Service :



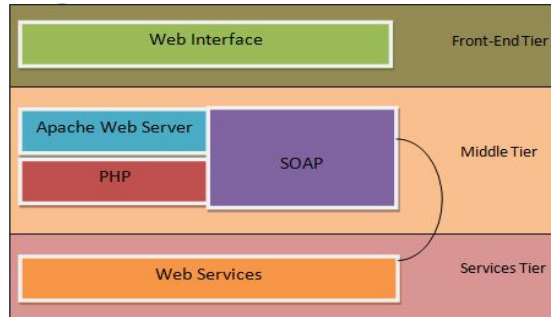
**Gambar 6. Skema Diagram WMS Dashboard (Bagian Service)**

#### 4. Layer Aplikasi WMS Dashboard

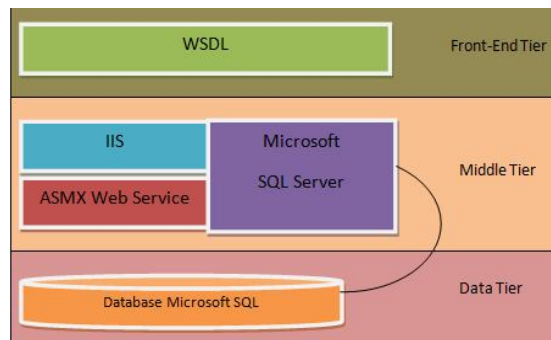
Layer aplikasi adalah stuserktur lapisan teknologi yang digunakan untuk membentuk aplikasi. Layer aplikasi *Dashboard* WMSterdiri dari 2 diagram layer yaitu diagram untuk server dan diagram untuk client, dimana setiap diagram terdiri dari 3 buah lapisan. Arsitektur layer aplikasi *Dashboard* WMS digambarkan sebagai berikut :

Layer aplikasi adalah stuserktur lapisan teknologi yang digunakan untuk membentuk aplikasi. Layer aplikasi *Dashboard* WMSterdiri dari 2 diagram layer yaitu diagram untuk server dan diagram untuk client, dimana setiap diagram terdiri dari 3 buah lapisan. Arsitektur layer aplikasi *Dashboard* WMS digambarkan pada gambar 7 dan 8 sebagai berikut :





**Gambar 7. Diagram layer di sisi client**

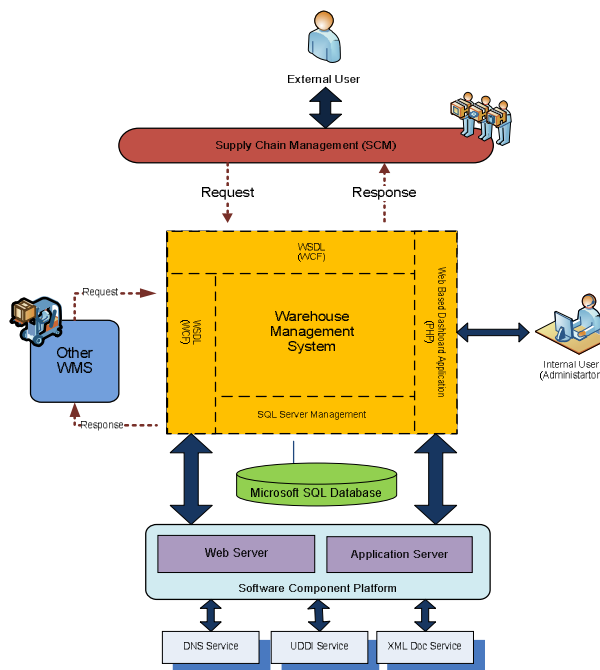


**Gambar 8. Diagram layer di sisi server**

#### 5. Arsitektur Aplikasi WMS *Dashboard* dengan User dan System Lain

Aplikasi WMS *Dashboard* selain dapat langsung digunakan oleh User, juga dapat digunakan oleh System lain. System lain memanfaatkan service yang dimiliki oleh WMS *Dashboard* dengan cara mengambil WSDL dari service yang dibutuhkannya. Pengambilan WSDL

tersebut dapat menggunakan protokol SOAP atau HTTP POST. Seangkan untuk user dapat mengakses aplikasi WMS *Dashboard* dengan memanfaatkan protokol HTTP dengan menggunakan PHP. Gambar 9 Berikut ini menjelaskan arsitektur WMS *Dashboard* dengan user dan sistem lainnya:



Gambar 9. Arsitektur WMS Dashboard dengan user dan sistem lain

6. WMS Service (System Interface)

Service untuk aplikasi WMS Dashboard dibagi menjadi 2 Service yaitu :

1. WMSDService

WMSD Service berisi service-service yang berhubungan dengan tabel-tabel data mentahan dari WMS. Service-service yang ada pada WMSD service adalah :

Tabel 3 . Tabel Service- service pada WMSDService

NAMA SERVICE	KEGUNAAN SERVICE
checkUser	Memeriksa user yang menggunakan Service dan menggunakan database
deleteCity	Menghapus data pada tabel City
deleteFinancialTransaction	Menghapus data pada tabel

NAMA SERVICE	KEGUNAAN SERVICE
	FinancialTransaction
deleteProvince	Menghapus data pada tabel province
deleteRack	Menghapus data pada tabel rack
deleteRole	Menghapus data pada tabel role
deleteStock	Menghapus data pada tabel stock
deleteStockType	Menghapus data pada tabel stocktype
deleteTransaction Report	Menghapus data pada tabel transactionreport
deleteUser	Menghapus data pada tabel user
deleteWarehouse	Menghapus data pada tabel warehouse
getCity	Mengambil data dari tabel city
getDescRole	Mengambil deskripsi dari tiap role pada tabel role
getFinancialTransaction	Mengambil data dari tabel financial transaction
getFinancialTran	Mengambil data dari

NAMA SERVICE	KEGUNAAN SERVICE
sactionType	tabel financialtransactiontype
getListCity	Mengambil data dari tabel city berupa list
getListFinancialTrans	Mengambil data dari tabel financial transaction berupa list
getListProvince	Mengambil data dari tabel province berupa list
getListRack	Mengambil data dari tabel rack berupa list
getListRole	Mengambil data dari tabel role berupa list
getListStock	Mengambil data dari tabel stock berupa list
getListStockType	Mengambil data dari tabel stocktype berupa list
getListTransactionReport	Mengambil data dari tabel transactionreport berupa list
getListUser	Mengambil data dari tabel user berupa list
getListWarehouse	Mengambil data dari tabel warehouse berupa list
getProvince	Mengambil data dari tabel province
getRack	Mengambil data dari tabel rack
getStock	Mengambil data dari tabel stock
getStockType	Mengambil data dari tabel stocktype
getTrans	Mengambil data dari tabel transaction
getTransactionType	Mengambil data dari tabel transactiontype
getUser	Mengambil data dari tabel user
getWarehouse	Mengambil data dari tabel warehouse
getusername	Mengambil data username dari tabel user
newCity	Menambah data pada tabel City
newFinancialTransaction	Menambah data pada tabel FinancialTransaction
newProvince	Menambah data pada

NAMA SERVICE	KEGUNAAN SERVICE
	tabel province
newRack	Menambah data pada tabel rack
newRole	Menambah data pada tabel role
newStock	Menambah data pada tabel stock
newTransactionReport	Menambah data pada tabel TransactionReport
newUser	Menambah data pada tabel User
newWarehouse	Menambah data pada tabel Warehouse
newstockType	Menambah data pada tabel Stooctype
updateCity	Mengubah data pada tabel City
updateFinancialTransaction	Mengubah data pada tabel Financial Transaction
updateProvince	Mengubah data pada tabel province
updateRack	Mengubah data pada tabel rack
updateRole	Mengubah data pada tabel role
updateStock	Mengubah data pada tabel stock
updateStockType	Mengubah data pada tabel stocktype
updateTransactionReport	Mengubah data pada tabel transactionreport
updateUser	Mengubah data pada tabel user
updateWarehouse	Mengubah data pada tabel warehouse

## 2. KPIProfit Service

KPIProfit Service berisi service-service yang berhubungan langsung dengan tabel KPI dan tabel Profit, dimana kedua tabel tersebut akan menjadi sumber dari pembuatan report. Servicenya antara lain:

**Tabel 4. Tabel Service- service pada KPIProfit Service**

NAMA SERVICE	KEGUNAAN SERVICE
getKPI	Mendapatkan data dari table KPI berupa list
getProfitReport	Mendapatkan data dari table profit Report berupa list

### 7. WMS Dashboard Application (User Interface)

User Interface dari aplikasi WMS Dashboard dibagi menjadi 3 bagian, yaitu User Interface untuk General User, SCM/Admin dan User/Owner.

#### 1. User Interface untuk General User/ Visitor

User Interface untuk Internal User/Admin terdiri dari 2 Halaman utama, yaitu :

- a. Halaman Login



**Gambar 10. Halaman Login**

- b. Halaman About WMS

#### 2. User Interface untuk Admin/ SCM

User Interface untuk Admin/ SCM terdiri dari 15 Halaman utama, yaitu :

- a. Halaman Home



**Gambar 11. Halaman Home Admin / SCM**

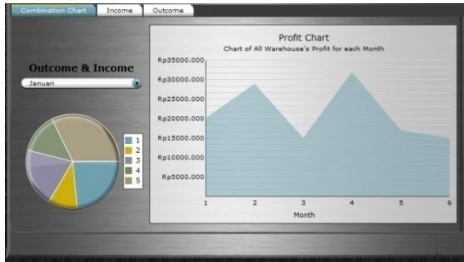
- b. Halaman Tabel User
- c. Halaman Tabel Role
- d. Halaman Tabel Rack
- e. Halaman Tabel Stock
- f. Halaman Tabel Stock Type
- g. Halaman Tabel Warehouse
- h. Halaman Tabel City
- i. Halaman Tabel Province
- j. Halaman Tabel Financial Transaction
- k. Halaman Tabel Financial Transaction Type
- l. Halaman Tabel Transaction
- m. Halaman Tabel Transaction Type
- n. Halaman Tabel KPI
- o. Halaman Tabel Profit

#### 3. User Interface untuk User/Owner

User Interface untuk User/Owner terdiri dari 8 Halaman utama, yaitu :

- a. Halaman Home
- b. Halaman Dashboard Stock Status
- c. Halaman Dashboard Warehouse Space Status
- d. Halaman Dashboard SOP Activity Status
- e. Halaman Dashboard Stock Percentage

- f. Halaman *Dashboard* Warehouse Space Percentage
- g. Halaman *Dashboard* SOP Activity Percentage
- h. Halaman *Dashboard* Profit



**Gambar 12. Halaman *Dashboard* Profit**

## D. Kesimpulan Dan Saran

### 1 Kesimpulan

Berdasarkan aplikasi yang telah dirancang, dapat disimpulkan bahwa:

1. *Web Service* untuk aplikasi WMS *Dashboard* memiliki total 56 *Service* yang mengacu kepada 14 tabel dari database WMS.
2. Aplikasi WMS *Dashboard* memiliki 2 menu untuk general user, 15 menu untuk SCM/admin dan 8 menu untuk owner/user.

### 2 Saran

Saran yang dapat disampaikan untuk aplikasi WMS *Dashboard* ini adalah:

1. *Web Service* WMS *Dashboard* hendaknya distandardisasikan dan didaftarkan pada UDDI, sehingga *service* tersebut dapat dimanfaatkan oleh masyarakat luas.

2. Untuk meningkatkan mobilitas dan update data secara otomatis WMS *dashboard* sebaiknya diintegrasikan dengan teknologi RFID .
3. Sekuritas *Web Service* hendaknya ditingkatkan dengan melakukan enkripsi pada XML yang ditransportasikan
4. Untuk manajemen loading data pada aplikasi WMS *dashboard* disisi client sebaiknya memanfaatkan frame atau teknologi JQuery sehingga lebih meng-efektifkan *loading time*, sehingga user tidak menunggu terlalu lama saat mengakses aplikasi WMS *Dashboard*.

## E. Daftar Pustaka

- Didiek S Wiyono.(2009),TESIS: Analisis dan Perancangan Aplikasi Web dan Mobile Supply Chain Management pada Distribusi Komoditas Padi Pascapanen (Studi Kasus Sistem SAPA Sukabumi),Institut TeknologiBandung
- Martin Verwijmeren (2004), Software component architecture in supply chain management, Journal Computers in Industry,53,pp. 165–178
- Steve Banker. (2006), WMS Overview, <http://www.frame-wx.com/eng/imgs/02.pdf> diakses pada tanggal 16 November 2010
- Kadir Abdul, 2004. Dasar Pemrograman Web Dinamis Menggunakan PHP. Yogyakarta : Andi
- Richard Robert. 2006. Pro PHP XML and Web Services. New York : Appress.
- Christian Nagel, et all. 2010. *Professional C# 4 and .NET 4*. Wiley Publishing

*Wikipedia.2011.Simple Object Access  
Protocol.*  
<http://id.wikipedia.org/wiki/SOAP>

*diakses pada tanggal 23 Februari  
2011*  
*Peter Lavin. 2006. Object-OrientedPHP .  
San Fransisco:No Starch Press*